Cloudera Streaming Analytics 1.14.0

Release Notes

Date published: 2019-12-17
Date modified: 2024-12-03

https://docs.cloudera.com/

Legal Notice

Contents

What's new in Cloudera Streaming Analytics

Cloudera Streaming Analytics 1.14.0 covers new features beside the core streaming functionality of Apache Flink and SQL Stream Builder.

Cloudera platform support

Cloudera Streaming Analytics 1.14.0 is supported as a Long Term Support (LTS) version on Cloudera on premises 7.3.1. Ensure that you review the 7.3.1 Release Notes and Support Matrix to understand which operating system, database, and JDK versions are supported for Cloudera Streaming Analytics as well.

KNOX for Cloudera SQL Stream Builder as Load-Balancer

From Cloudera Streaming Analytics 1.14.0, Cloudera SQL Stream Builder uses KNOX as a load-balancer if there are multiple instances.

The Load-balancer role, associated Cloudera Manager configurations, and nginx binaries have been removed.

Important: KNOX is the only load-balancing solution available for Cloudera SQL Stream Builder. If your cluster does not support KNOX, load-balancing will not be available for Cloudera SQL Stream Builder.

For more information, see Enabling High Availability for Cloudera SQL Stream Builder.

Note: KNOX requires Server-Sent Events (SSE) to be enabled in Cloudera Manager.

Cloudera SQL Stream Builder sampling now using the Server-Sent Events (SSE) protocol

From Cloudera Streaming Analytics 1.14.0, the sampling functionality in Cloudera SQL Stream Builder is using the Server-Sent Events (SSE) protocol to support using KNOX as a load-balancer.

New Cloudera Manager configurations and REST API endpoints were added for SSE sampling.

Important: SSE support is disabled by default. To use it, you have to enable it in Cloudera Manager, see Enabling High Availability for Cloudera SQL Stream Builder.

Fixed issues

Review the list of Flink and Cloudera SQL Stream Builder issues that are resolved in Cloudera Streaming Analytics 1.14.0.

CSA-5423 - Extend Cloudera SQL Stream Builder diag bundle data points

CSA-5440 - Permit Spring Flyway plugin execution on PvC

CSA-5364 - Add number of topics/tables to successful data source validation message on UI

CSA-5306 - Cloudera SQL Stream Builder API does not validate catalog type

CSA-5362 - Update "ssb-sse" ASCII text banner to not contain special characters

CSA-5359 - Improve error message when creating a JS UDF with a Java version that doesn't support it

CSA-5296 - Samples table fields are limited to 32 characters in mysql and oracle dbs

CSA-5324 - Cloudera SQL Stream Builder default admin does not have admin privileges

CSA-5428 - Sampling renders null as Invalid Number in some cases

CSA-5474 - Cloudera SQL Stream Builder can't execute any jobs due to permission issue in the SSB artifacts directory

CSA-5475 - Local-kafka connector template not showing in Cloudera SQL Stream Builder

CSA-5479 - Using Temp View based on kudu lookup table leaks eventpolls

CSA-5499 - Bump Avro to 1.11.4 in parcel to mitigate CVE-2024-47561

Deprecation notices in Cloudera Streaming Analytics 1.14.0

Certain features and functionalities have been removed or deprecated in Cloudera Streaming Analytics 1.14.0. You must review these items to understand whether you must modify your existing configuration. You can also learn about the features that will be removed or deprecated in the future release to plan for the required changes.

Terminology

Items in this section are designated as follows:

Deprecated

Technology that Cloudera is removing in a future Cloudera Streaming Analytics release. Marking an item as deprecated gives you time to plan for removal in a future Cloudera Streaming Analytics release.

Moving

Technology that Cloudera is moving from a future Cloudera Streaming Analytics release and is making available through an alternative Cloudera offering or subscription. Marking an item as moving gives you time to plan for removal in a future Cloudera Streaming Analytics release and plan for the alternative Cloudera offering or subscription for the technology.

Removed

Technology that Cloudera has removed from Cloudera Streaming Analytics and is no longer available or supported as of this release. Take note of technology marked as removed since it can potentially affect your upgrade plans.

Deprecation Notices for Cloudera SQL Stream Builder

Certain features and functionality are deprecated or removed in Cloudera SQL Stream Builder. You must review these changes along with the information about the features in Cloudera SQL Stream Builder that will be removed or deprecated in a future release.

Deprecated

v1 REST API

The v1 REST API for Cloudera SQL Stream Builder has been deprecated and will be removed in a future version of Cloudera Streaming Analytics.

Customers are advised to migrate to the v2 API, available for Cloudera SQL Stream Builder.

For more information on the v2 API, see Cloudera SQL Stream Builder REST API reference.

Support for JavaScript UDFs

Due to the deprecation of the Nashorn engine used in JDK 8 and 11, User-Defined Functions (UDFs) written in JavaScript has been deprecated in Cloudera Streaming Analytics 1.13.0. Cloudera

recommends that customers start using Python UDFs for all new developments, and start migrating their JavaScript UDFs to Python to prepare for future upgrades.

Known issues and limitations

Learn about the known issues in Flink and Cloudera SQL Stream Builder, the impact or changes to the functionality, and the workaround in Cloudera Streaming Analytics 1.14.0.

Cloudera SQL Stream Builder

CSA-5747 - Upgrade causes attribute conversion failure when there are data sources containing secret properties

Due to the secret property encryption for data sources introduced in Cloudera Streaming Analytics 1.14.0, after upgrading from a version without the encryption (1.13.x or lower), data sources with secret properties may cause Cloudera Streaming Analytics not starting, or the Explorer Cloudera SQL Stream Builder failing to load resources properly.

1. Connect to the configured Cloudera SQL Stream Builder admin database. The database information can be found in Cloudera ManagerSQL Stream Builder ServiceConfiguration and filtering for database.
2. In the database, select the records of the data_sources table.
3. Remove all data_source entries that has properties containing keys matching the following keywords: secret, password, pwd, credentials, token, user-info, user.info.
4. Verify that the Cloudera SQL Stream Builder loads the resources properly.
5. Re-register the previously deleted data sources.

Note: Cloudera recommends that users delete data sources containing secrets **before upgrading** Cloudera Streaming Analytics, and re-register them after the upgrade is completed.

CSA-5564 - Unlocking keytab may fail in Cloudera SQL Stream Builder

When the user's credential salt in FreeIPA contains a double quote, unlocking the keytab in Cloudera SQL Stream Builder fails with an Invalid credentials error.

Reset/update the user's password. This triggers a regenerate action for the credential salt.

Note: Because the credential salt is generated automatically, it is possible that the new salt will also contain the offending double quote. In such cases, reset/update the password again.

ENGESC-23078 - Job not found after successful job creation

After successfully creating a job in Cloudera SQL Stream Builder, the SQL job is not found due to tables having empty values. This issue is indicated with the following error message in the log files:

```
java.lang.IllegalArgumentException: argument "content" is null
```

The issue only applies when upgrading from a Cloudera Streaming Analytics version lower than 1.9.0.

Update the empty values with null string in the mv_config and checkpoint_config fields as shown in the following example:

```
UPDATE jobs SET mv_config = 'null' WHERE mv_config IS NULL;
UPDATE jobs SET checkpoint_config = 'null' WHERE checkpoint_conf
ig IS NULL;
```

CSA-4858 - Kerberos encryption type detection does not always work correctly for Cloudera SQL Stream Builder

Cloudera SQL Stream Builder detects no supported encryption types even though there is a list of allowed encryption types in the krb5.conf file. This causes an error when generating keytabs from the principal and password pair.

1. Run ktutil on your cluster.
2. Change the configuration with the following commands:

```
addent -password -p [***USERNAME***] -k 1 -e aes256-cts
wkt /tmp/new_keytab.keytab
```

3. Upload the new keytab to Cloudera SQL Stream Builder.

Auto discovery is not supported for Apache Knox

You need to manually configure Knox with Cloudera SQL Stream Builder to enable Knox authentication. Complete the configuration based on the Cloudera Base on premises version you use. For more information, see the Enabling Knox authentication for Cloudera SQL Stream Builder documentation.

Flink

CSA-5525 - Illegal join reordering in Flink optimizer

Flink optimizer's reordering might violate certain clauses (for example FOR SYSTEM_TIME AS OF) that are supported only on a specific side of a join operation, resulting in an error.

Example error message:

```
Caused by: org.apache.flink.table.api.TableException: Temporal t
able join only support apply FOR SYSTEM_TIME AS OF on the right
table
```

Set table.optimizer.join-reorder-enabled to false, until the issue is fixed in upstream Flink.

DataStream conversion limitations

- Converting between Tables and POJO DataStreams is currently not supported in Cloudera Streaming Analytics.
- Object arrays are not supported for Tuple conversion.
- The java.time class conversions for Tuple DataStreams are only supported by using explicit TypeInformation: LegacyInstantTypeInfo, LocalTimeTypeInfo.getInfoFor(LocalDate/LocalDateTime/LocalTime.class).
- Only java.sql.Timestamp is supported for rowtime conversion, java.time.LocalDateTime is not supported.

Kudu catalog limitations

- CREATE TABLE
  - Primary keys can only be set by the kudu.primary-key-columns property. Using the PRIMARY KEY constraint is not yet possible.
  - Range partitioning is not supported.
- When getting a table through the catalog, NOT NULL and PRIMARY KEY constraints are ignored. All columns are described as being nullable, and not being primary keys.
- Kudu tables cannot be altered through the catalog other than simply renaming them.

Schema Registry catalog limitations

- Currently, the Schema Registry catalog / format only supports reading messages with the latest enabled schema for any given Kafka topic at the time when the SQL query was compiled.
- No time-column and watermark support for Registry tables.
- No CREATE TABLE support. Schemas have to be registered directly in the SchemaRegistry to be accessible through the catalog.

- The catalog is read-only. It does not support table deletions or modifications.
- By default, it is assumed that Kafka message values contain the schema id as a prefix, because this is the default behaviour for the SchemaRegistry Kafka producer format. To consume messages with schema written in the header, the following property must be set for the Registry client: store.schema.version.id.in.header: true.

Behavioral changes

Learn about the change in certain functionality of Flink and Cloudera SQL Stream Builder that has resulted in a change in behavior from the previously released version of Cloudera Streaming Analytics.

Summary:

Use KNOX for Cloudera SQL Stream Builder as Load-Balancer

Previous behavior:

Cloudera SQL Stream Builder used a Load Balancer instance with nginx to support high availability.

New behavior:

Cloudera SQL Stream Builder only uses KNOX as a load-balancer if there are multiple instances. See Enabling High Availability for Cloudera SQL Stream Builder.

Summary:

Cloudera SQL Stream Builder sampling now uses the Server-Sent Events (SSE) protocol.

Previous behavior:

Cloudera SQL Stream Builder samples were delivered via a websocket connection.

New behavior:

The sampling functionality in Cloudera SQL Stream Builder is using the Server-Sent Events (SSE) protocol to support using KNOX as a load-balancer. See Enabling High Availability for Cloudera SQL Stream Builder.

Unsupported features

Some Apache Flink and Cloudera SQL Stream Builder features exist in Cloudera Streaming Analytics 1.14.0, but are not supported by Cloudera. These features are not ready for production deployment, but Cloudera encourages you to explore them in non-production environments and provide feedback on your experiences through the Cloudera Community Forums.

Cloudera SQL Stream Builder

- Virtual environments for Python are not supported.
- Direct Cloudera SQL Stream Builder upgrade from 1.3.0

Important: This does not impact Flink, you can directly upgrade Flink as described in the documentation.

For more information, see the Upgrading SQL Stream Builder in the 1.3.0 documentation.

Flink

- Apache Flink batch (DataSet) API
- GPU Resource Plugin
- SQL Client

- The following features are not supported in SQL and Table API:
  - HBase Table Connector
  - Old Planner
  - Non-windowed (unbounded) joins, distinct

Support Matrix

Before installing Cloudera Streaming Analytics, review the supported components, databases, connectors and the default ports in use for Flink and Cloudera SQL Stream Builder.

Component support

Learn more about which Apache Flink component version is supported in the Cloudera Streaming Analytics releases.

Cloudera Streaming Analytics version	Component version
Cloudera Streaming Analytics 1.14.0	Apache Flink 1.19.1
Cloudera Streaming Analytics 1.13.0	Apache Flink 1.19.1
Cloudera Streaming Analytics 1.12.0	Apache Flink 1.18.0
Cloudera Streaming Analytics 1.11.0	Apache Flink 1.16.2
Cloudera Streaming Analytics 1.10	Apache Flink 1.16.1
Cloudera Streaming Analytics 1.9.0	Apache Flink 1.15.1
Cloudera Streaming Analytics 1.8.0	
Cloudera Streaming Analytics 1.7.0	Apache Flink 1.14
Cloudera Streaming Analytics 1.6.2	
Cloudera Streaming Analytics 1.6.1	
Cloudera Streaming Analytics 1.6.0	
Cloudera Streaming Analytics 1.5.3	Apache Flink 1.13
Cloudera Streaming Analytics 1.5.1	
Cloudera Streaming Analytics 1.5.0	
Cloudera Streaming Analytics 1.4.1	Apache Flink 1.12
Cloudera Streaming Analytics 1.4.0	
Cloudera Streaming Analytics 1.3.0	
Cloudera Streaming Analytics 1.2.0	Apache Flink 1.10
Cloudera Streaming Analytics 1.1.0	Apache Flink 1.9.1

Related Information

Cloudera Streaming Analytics 1.14.0 Release Notes
Cloudera Streaming Analytics 1.13.2 Release Notes
Cloudera Streaming Analytics 1.13.1 Release Notes
Cloudera Streaming Analytics 1.13.0 Release Notes
Cloudera Streaming Analytics 1.12.0 Release Notes
Cloudera Streaming Analytics 1.11.0 Release Notes
Cloudera Streaming Analytics 1.10.0 Release Notes
Cloudera Streaming Analytics 1.9.0 Release Notes

Cloudera Streaming Analytics 1.8.0 Release Notes
Cloudera Streaming Analytics 1.7.0 Release Notes
Cloudera Streaming Analytics 1.6.2 Release Notes
Cloudera Streaming Analytics 1.6.1 Release Notes
Cloudera Streaming Analytics 1.6.0 Release Notes
Cloudera Streaming Analytics 1.5.3 Release Notes
Cloudera Streaming Analytics 1.5.1 Release Notes
Cloudera Streaming Analytics 1.5.0 Release Notes
Cloudera Streaming Analytics 1.4.1 Release Notes
Cloudera Streaming Analytics 1.4.0 Release Notes
Cloudera Streaming Analytics 1.3.0 Release Notes
Cloudera Streaming Analytics 1.2.0 Release Notes
Cloudera Streaming Analytics 1.1.0 Release Notes

System Requirements

Before installing Cloudera Streaming Analytics, you should verify that you meet the system requirements. Other than Cloudera Base on premises, you should also check the latest supported version of the needed components.

For detailed information about the supported versions of Cloudera Base on premises, operating systems and databases, see the Cloudera Support Matrix.

Apache Flink support	**1.19.1**
Cloudera Runtime component support in Cloudera Base on premises	**7.3.1**
Atlas	2.1.0
HBase	2.4.17
HDFS	3.1.1
Hive	3.1.3
Kafka[1]	3.4.1
Kudu	1.17.0
Schema Registry	0.10.0
Streams Messaging Manager	2.3.0
Apache Iceberg	1.3.1

Connector support	
JDBC PostgreSQL	9.6-16
JDBC MySQL	5.7, 8
JDBC Hive	3.1.3
JDBC Oracle	19, 19c, 21c, 23c
JDBC Db2	11.5
JDBC SQL Server	2007-2022
CDC PostgreSQL	9.6-16
CDC MySQL	5.7, 8
CDC Oracle	19, 19c, 21c, 23c

[1] Connecting to Kafka that is running on remote CDH6 or HDP3 is also supported.

Connector support	
CDC Db2	11.5
CDC SQL Server	2007-2022
Apache Iceberg	1.3.1

Default ports for Flink and Cloudera SQL Stream Builder

You need to use the default ports of Flink and Cloudera SQL Stream Builder when you need to reach or connect to their services. The default port are set in Cloudera Manager, but can be changed if required.

The following table lists the default ports and the corresponding property file names for Flink and Cloudera SQL Stream Builder. The ports are set by default in Cloudera Manager. You can change the ports as required using the configuration properties.

Component	Service	Port	Configuration property
Flink	Flink Dashboard	18211	historyserver.web.port
Cloudera SQL Stream Builder	Streaming SQL Engine	18121	server.port
	Materialized View Engine	18131	server.port
Cloudera SQL Stream Builder with Load Balancer	Streaming SQL Engine	8080	ssb.sse.loadbalancer.server.port
	Secured Streaming SQL Engine	8445	ssb.sse.loadbalancer.server.secure.port
	Materialized View Engine	8081	ssb.mve.loadbalancer.server.port
	Secured Materialized View Engine	8444	ssb.mve.loadbalancer.server.secure.port

For the default port list of the Cloudera Runtime components, see the *Ports Used by Cloudera Runtime Components* document.

Maven dependencies in Flink

Review the list of Maven dependencies to ensure the correct connector versions in your Flink applications.

Avro

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-avro</artifactId>
  <version>1.19.1-csa1.14.0.0</version>
</dependency>
```

Confluent Registry

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-confluent-registry</artifactId>
  <version>1.0-csa1.14.0.0</version>
</dependency>
```

CSV

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-csv</artifactId>
  <version>1.19.1-csa1.14.0.0</version>
```

```
</dependency>
```

Hive

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-hive_2.12</artifactId>
  <version>1.19.1-csa1.14.0.0</version>
</dependency>
```

HBase

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-hbase-1.4</artifactId>
  <version>3.0-csa1.14.0.0</version>
</dependency>
```

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-hbase-2.4</artifactId>
  <version>3.0-csa1.14.0.0</version>
</dependency>
```

Iceberg

```
<dependency>
  <groupId>org.apache.iceberg</groupId>
  <artifactId>iceberg-flink-runtime-1.17</artifactId>
  <version>1.3.1.7.3.1.0-197</version>
</dependency>
```

JDBC

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-jdbc</artifactId>
  <version>3.2-csa1.14.0.0</version>
</dependency>
```

JSON

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-json</artifactId>
  <version>1.19.1-csa1.14.0.0</version>
</dependency>
```

Kafka

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-kafka</artifactId>
  <version>3.2-csa1.14.0.0</version>
</dependency>
```

Kudu

```
<dependency>
  <groupId>org.apache.bahir</groupId>
```

```
  <artifactId>flink-connector-kudu_2.12</artifactId>
  <version>1.1.0-csa1.14.0.0</version>
</dependency>
```

Schema Registry

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-cloudera-registry</artifactId>
  <version>1.0-csa1.14.0.0</version>
</dependency>
```

Table API

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-table-api-java-bridge</artifactId>
  <version>1.19.1-csa1.14.0.0</version>
</dependency>
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-table-planner-loader-bundle</artifactId>
  <version>1.19.1-csa1.14.0.0</version>
</dependency>
```

For more information about how to use Maven in Flink, see the Apache documentation.

Flink API Support

Cloudera Streaming Analytics offers support for three fundamental layers of the Apache Flink API. You can use DataStream API, the ProcessFunction API and a selected subset of the SQL API to develop your Flink streaming applications.

From the DataStream and ProcessFunction APIs, the following are supported based on the support annotations provided by the Apache Flink community.

Stable (@Public)	Evolving (@PublicEvolving)
• DataStream API	• ProcessFunction • Stream Join • Interval Join • Stateful operators • FsStatebackend with HDFS • RocksDBStateBackend with HDFS

Note: Cloudera Streaming Analytics does not support batch processing (DataSet API).